

High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation

Eloi Ruiz-Gironés^{a,*}, Xevi Roca^b, Jose Sarrate^a

^a *Laboratori de Càlcul Numèric (LaCàN), Departament de Matemàtica Aplicada III (MA III), Universitat Politècnica de Catalunya (UPC), Campus Nord UPC, 08034 Barcelona, Spain.*

^b *Aerospace Computational Design Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.*

Abstract

We propose a 3D mesh curving method that converts a straight-sided mesh to an optimal-quality curved high-order mesh that interpolates a CAD boundary representation. The main application of this method is the generation of discrete approximations of curved domains that are valid for simulation analysis with unstructured high-order methods. We devise the method as follows. First, the boundary of a straight-sided high-order mesh is curved to match the curves and surfaces of a CAD model. Second, the method minimizes the volume mesh distortion with respect to the coordinates of the inner nodes and the parametric coordinates of the curve and surface nodes. The proposed minimization features untangling capabilities and therefore, it repairs the invalid elements that may arise from the initial curving step. Compared with other mesh curving methods, the only goal of the proposed residual system is to minimize the volume mesh distortion. Furthermore, it is less constrained since the boundary nodes are free to slide on the CAD curves and surfaces. Hence, the proposed method is well suited to generate curved high-order meshes of optimal quality from CAD models that contain thin parts or high-curvature entities. To illustrate these capabilities, we generate several curved high-order meshes from CAD models with the implementation detailed in this work. Specifically, we detail a node-by-node non-linear iterative solver that minimizes the proposed objective function in a block Gauss-Seidel manner.

Keywords: Curved high-order meshing, mesh smoothing, mesh untangling, mesh quality, CAD interpolation, hierarchical minimization

1. Introduction

In the last decade, unstructured high-order methods [1–5] have been proved to be well suited to perform high-fidelity simulations on complex domains. First, they are ready to approximate complex curved domains. Second, they feature low numerical dissipation and dispersion. Third, they are faster than low-order methods in several applications [6–12], specially for problems that require high-accuracy and implicit time stepping. Finally, they favor dense floating-point operations that can

be efficiently implemented in current computing architectures. A key ingredient to meet all these practical advantages is the generation of curved high-order discretizations of the simulation domain.

In the most common framework, curved high-order meshes are generated using an *a posteriori* approach referred as mesh curving [13–22]. First, an initial linear mesh that features the element size and stretching required by the simulation is generated. For instance, the mesh may present higher resolution in the regions of interest, and highly stretched elements to capture the boundary layers or shocks in the solution. Then, the linear mesh is converted to a straight-sided high-order mesh by introducing additional nodes. Finally, the straight-sided high-order mesh is curved to match the corresponding CAD model boundary.

*Corresponding author. Tel.: +34 93 401 18 42 ; fax: +34 93 401 1825

Email addresses: eloi.ruiz@upc.edu (Eloi Ruiz-Gironés), xeviroca@mit.edu (Xevi Roca), jose.sarrate@upc.edu (Jose Sarrate)

Typically, the mesh curving is performed in two steps. First, the high-order nodes at the mesh boundary are relocated on the CAD boundary entities to ensure that the corresponding surface elements are smooth and valid. This curving of the mesh boundary may introduce inverted volume elements. Then, the boundary nodes are fixed and the inner high-order nodes are relocated (smoothing) to accommodate the curved boundary and repair the invalid elements (untangling). However, the fixed coordinates of the boundary nodes may excessively constrain the volume mesh and therefore, low-quality or inverted volume elements could be obtained for specific configurations, specially for those configurations that present high-curvature entities, thin regions, or opposite curves and surfaces meshed with the same topology of structured elements.

To overcome this issue, we propose an untangling and smoothing process based on minimizing a global objective function defined only in terms of the distortion of the volume mesh [23], that takes the coordinates of the high-order nodes as arguments [22, 24–26]. The novelty of the approach is that we consider a single target objective function (mesh distortion) where all the nodes, except the nodes located on the geometry vertices, are free to move on the corresponding container CAD entity. Thus, we directly improve the quality of the volume mesh in a single step, instead of first relocating the boundary elements, according to a given surface curving criteria, and then improving the inner elements, according to a different volume curving criteria.

It is important to point out that our method targets the quality of the volume mesh, which is critical in the simulation analysis. That is, low-quality meshes may hinder the accuracy of the simulation approximation to the solution. Furthermore, if the quality of the volume mesh is optimal, then the quality of the boundary mesh is high. Counter-wise, optimal surface meshes can determine the boundary of low-quality or even invalid volume meshes. That is, the reciprocal, the case for standard mesh curving methods, is not true in general.

Although the objective function is globally defined for all the nodes, we propose to optimize it using a hierarchical approach in three stages. First, we improve the quality of the volume mesh by moving the curve nodes along the CAD curves. Second, we improve the quality of the volume mesh by moving the surface nodes on the CAD surfaces. Finally,

we improve the quality of the volume mesh moving the interior nodes. During the first two steps of the proposed method, we have to ensure that the boundary nodes are located on the corresponding curves and surfaces. To this end, we consider the parameterization of curves and surfaces provided by the CAD model. Specifically, the unknowns that correspond to the curve and surface nodes are determined by their parametric coordinates on the CAD entities as in [27, 28]. Following this approach, we ensure that the nodes lie on the corresponding entity, and we avoid expensive and non-robust projections of the nodes onto the boundary.

The main advantage of the proposed approach is that we only take into account the quality of the volume mesh. Moreover, the location of the curve and surface nodes is not fixed. This results in an optimization process that is less constrained than the standard approach (fixed boundary nodes) and therefore, a better configuration of nodes can be achieved. It is important to point out that the node type (vertex, curve, surface or volume) is assigned during the hierarchical meshing process. That is, when the curve, surface and volume meshes are generated, the corresponding nodes are marked with the appropriate type.

The remaining of the paper is structured as follows. In Section 2, we review the existing literature related to the presented work. In Section 3, we present a simple two-dimensional example to illustrate the objectives of this work. In Section 4 we present the scope of this work, the statement of the problem we aim to solve, and the selected approach. In Section 5, we show the proposed formulation for the optimization process. In Section 6, we present the proposed hierarchical optimization approach. Then, in Section 7, we present several examples in order to show that the proposed method is able to untangle inverted elements and improve the overall mesh quality. Finally, in Section 8, we present the concluding remarks and the future work.

2. Related work

References [29–31], introduce a framework to define element quality and distortion measures for linear elements. These measures can be written in terms of the Jacobian of the mapping between an ideal and a physical element. Thereafter, these measures were extended to quantify the quality of high-order elements [18, 22, 24–26]. Using these

algebraic distortion measures, several authors proposed to perform an optimization process to compute the optimal node position that minimize the mesh distortion, see [31–35] for linear elements and [22, 24–26] for high-order elements. The minimization process can be performed globally or locally. In the global approach, an objective function has to be minimized for all the nodes at the same time. In the local approach, all the free nodes are smoothed sequentially until the objective function is optimized.

The objective functions introduced by Knupp present asymptotes (barriers) when inverted elements appear in the mesh. For this reason, when tangled elements are present, the optimization process is not able to recover a valid configuration of the nodes to obtain untangled elements. To overcome this drawback, in [31, 34], a two-stage smoothing process is proposed. In the first stage, an objective function to untangle the inverted elements is used. In the second stage, a different objective function to improve the quality is used. That is, the optimization process is driven by two objective functions: the first one to untangle the mesh, and the second one to smooth the mesh. Other authors used the log-barrier method in order to avoid tangled elements, see [21, 36, 37]. A different approach to avoid the singularities, is to consider a regularized version of the distortion of the linear volume (boundary) elements surrounding an inner (boundary) node. In this manner, specific-purpose untangling and smoothing techniques using the shape distortion measure for triangular [38] and tetrahedral [39] meshes have been proposed. Note that we also consider the same regularization of the shape distortion measure. However, in our method the shape distortion measure is evaluated in a point-wise manner instead of considering the elements surrounding a node. Specifically, we consider a single target objective function that corresponds to the squared norm of the shape distortion measure of the whole high-order volume mesh.

Several algorithms deal with mesh smoothing on parameterized surfaces. These algorithms could be classified as indirect or direct. The indirect algorithms first compute the ideal position of a node and then they project back the node to the surface, see [38, 40–43]. On the other hand, direct methods express the objective function in terms of the parametric coordinates of the nodes [24, 44–47]. Thus, there is no need to project the nodes back to the surface. Note that these techniques are typically used during the process of obtaining a valid volume

mesh. First, a smoothing of the boundary mesh is carried out to obtain valid boundary elements and then, a smoothing of the interior nodes is performed to improve the quality of the whole mesh. Note that two different objective functions are used: one for the boundary nodes, and a different one for the interior nodes. The former is expressed in terms of the distortion of the boundary surface mesh, and the latter is expressed in terms of the distortion of the volume mesh. On the contrary, we consider the same objective function for the whole smoothing process. Specifically, our objective function has a single target that measures the distortion of the volume mesh. This single target formulation allows untangling and smoothing the whole mesh by moving at the same time the volume and surface nodes instead of using two specific-purpose stages for surface and volume meshes.

Different methods have been proposed to improve the quality of the volume mesh by moving the inner and boundary nodes. Note that the main issue is to impose that the boundary nodes should approximate the geometry boundary. To this end, it is possible to include a penalty term in the objective function that enforces that the corresponding nodes move close to the boundary [48]. It is also possible to impose that the nodes are strictly on the geometry boundary. In [49] the authors proposed a mesh quality improvement method that relocates the volume and surface nodes, but not the curve nodes. In some applications, such as curved high-order mesh generation or structured meshing, it could be also required to move the curve nodes. In [21] a method to smooth and untangle curved high-order meshes using the parametric coordinates of the surface and curve nodes was proposed. Specifically, they minimize an objective function that penalizes both large deformations and small values of the determinant of the iso-parametric mapping. In reference [50], the authors showed a smoothing method to increase the quality of iso-geometric meshes by moving the nodes on curves and surfaces. In our work, we also propose to move the nodes on the curves and surfaces based on their parametric coordinates. The difference here is that our goal is to minimize a regularized distortion measure of the volume mesh.

3. Motivation

To highlight the importance of moving the boundary nodes during an untangling and smoothing process, we show a simple linear quadrilateral

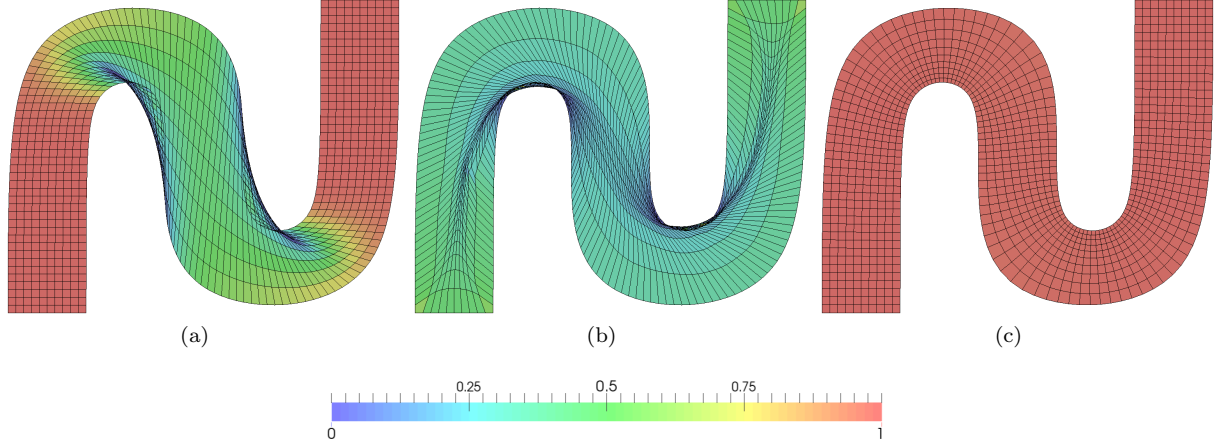


Figure 1: Simple quadrilateral mesh generated using the submapping method: (a) original mesh; (b) smoothing interior nodes; and (c) smoothing interior and boundary nodes.

mesh for a test geometry. Figure 1(a) presents the initial mesh generated using the submapping method. Note that the initial geometry is composed of four curves. Due to the curve curvature, when the boundary curves are discretized using intervals of the same length, the inner mesh contains inverted elements. When we apply a smoothing technique to improve the mesh quality, the method is not able to obtain a valid mesh even when the boundary is discretized using high-quality elements, see Figure 1(b). The objective of this work is to develop a smoother that is able to move the interior nodes as well as the boundary nodes according to the distortion measure of the volume elements. Using this procedure, a high-quality mesh is obtained, even when the boundary discretization constrains the quality of the whole mesh, see Figure 1(c).

4. Problem statement and methodology

4.1. Input and output

Our input data is a linear mesh, \mathcal{M}_1 , with the boundary nodes located on the corresponding CAD entities. We assume that the linear mesh is valid and its elements have the correct size and shape. We also assume that the entities of the geometry boundary, Σ_j , for $j = 1, \dots, N$ are parameterized by a continuously differentiable and invertible mapping (diffeomorphism) in such a way that:

$$\begin{aligned} \varphi_j : \mathcal{U}_j \subset \mathbb{R}^{d_j} &\longrightarrow \Sigma_j \subset \mathbb{R}^3 \\ \mathbf{u} = (u_1, \dots, u_{d_j}) &\longmapsto \mathbf{x} = \varphi_j(\mathbf{u}) \end{aligned}$$

where φ_j and d_j are the parameterization and the dimension of entity Σ_j , and \mathcal{U}_j is the parametric space. In our applications, we use the OpenCASCADE library [51] to obtain the parameterizations of the geometrical entities.

The output is a high-order mesh, \mathcal{M}_p , of polynomial degree p with all the boundary nodes located on the corresponding CAD entity, and composed of valid elements that have a shape and size close to the corresponding element in the initial linear mesh, \mathcal{M}_1 .

4.2. Methodology

The proposed approach is composed by the following four steps.

1. *Generate a linear mesh of the geometry.* Using any established linear mesh generator, we create a mesh with elements of the desired shape and size. Note that our approach requires the physical coordinates of all the nodes, and the parametric coordinates of the boundary nodes on the geometrical entity where they are located. This information is assigned while the mesh is generated hierarchically. That is, when the curve, surface and volume meshes are generated, the nodes store the entity where they belong to.
2. *Set the ideal mesh.* We increase the polynomial degree of the initial linear mesh on the physical space and we set this straight-sided high-order mesh as the ideal configuration in our optimization procedure. Note that this mesh is of the de-

sired polynomial degree and, at the same time, has elements of the desired size and shape.

3. *Set the initial curved high-order mesh.* We define a distribution of nodes of degree p on the straight-sided elements. For elements adjacent to the boundary, we bend the element edges and faces to match the geometry. Specifically, we use the Legendre-Gauss-Lobatto distribution of nodes [52] along the edges of the mesh. Then, the inner nodes of the mesh faces and cells are redistributed by means of the blending presented in [52]. Note that the boundary nodes are created on the parametric space of all the curves and surfaces that contain each node. Thus, we generate a mesh with the boundary nodes on the corresponding CAD entities and, for this reason, the mesh interpolates the geometry. The high-order meshes obtained after these operations may contain tangled elements, specially if high-curvature curves or surfaces are present in the geometry. These inverted elements appear due to two main issues. First, the *a posteriori* curving of the elements to match the geometry boundary can lead to element auto-intersections. Second, a valid-high-order distribution of nodes in the parametric space can be invalid once it is mapped into the physical space due to low-quality parameterizations of the boundary.
4. *Obtain a valid curved configuration of the high-order mesh.* We optimize the location of the nodes in order to minimize the distortion of the high-order volume mesh. We point out that the boundary nodes are moved on the parametric space of the geometrical entity where they are located. Thus, the position in the physical space of the boundary nodes is defined using the parameterization of its corresponding geometrical entity.

5. Formulation of the optimization process

5.1. Algebraic distortion measures for linear elements

According to [29, 30], the shape distortion measure for a linear element is

$$\eta(\mathbf{D}\phi(\mathbf{y})) = \frac{|\mathbf{D}\phi(\mathbf{y})|^2}{n\sigma(\mathbf{D}\phi(\mathbf{y}))^{2/n}}, \quad (1)$$

where \mathbf{y} is a point of the ideal element, ϕ is the mapping between an ideal element and the physical

element, $\mathbf{D}\phi$ is the Jacobian of ϕ , $\sigma(\cdot)$ is the determinant, n is the space dimension and $|\cdot| = \sqrt{(\cdot, \cdot)}$ is the Frobenius norm, being (\cdot, \cdot) a dot product for matrices, defined as

$$(\mathbf{A}, \mathbf{B}) = \text{tr}(\mathbf{A}^T \mathbf{B}).$$

We note that this distortion measure is invariant to translation and rotations, and equals 1 when the two elements only differ by a scale factor, and tends to ∞ when e_P becomes degenerate. The distortion measure can be used to improve the mesh quality by means of a minimization process, see [31–35]. However, the shape distortion measure presents asymptotes when $\sigma(\mathbf{D}\phi(\mathbf{y})) = 0$. This prevents the use of this distortion measure in a continuous optimization procedure. To overcome this drawback, in reference [39] a regularization of Equation (4) is introduced as

$$\eta_\delta(\mathbf{D}\phi(\mathbf{y})) = \frac{|\mathbf{D}\phi(\mathbf{y})|^2}{n\sigma_\delta(\mathbf{D}\phi(\mathbf{y}))^{2/n}}, \quad (2)$$

where

$$\sigma_\delta(\mathbf{D}\phi(\mathbf{y})) = \frac{1}{2} \left(\sigma + \sqrt{\sigma^2 + 4\delta^2} \right). \quad (3)$$

The δ parameter is defined as a small parameter, and its value depends on the problem. For a further analysis on the selection of δ for high-order meshes, see [22, 24, 47, 53]. In the case of linear meshes, the δ parameter can be chosen according to [38, 39, 54]

5.2. Continuous framework

Given an initial domain, Ω_I , we want to characterize a physical domain, Ω_P , in terms of a diffeomorphism $\phi^* \in \mathcal{C}^1(\Omega_I, \Omega_P)$. To determine the desired diffeomorphism ϕ^* , we consider to use a distortion measure [22] of mapping ϕ defined in terms of the regularized point-wise shape distortion measure [29, 30]. That is,

$$M\phi(\mathbf{y}) = \eta_\delta(\mathbf{D}\phi(\mathbf{y})). \quad (4)$$

In contrast with [22], where the optimal mapping ϕ^* is prescribed at boundary of the physical domain (i.e. the boundary mesh is fixed), we allow different diffeomorphisms candidates ϕ as long as $\phi(\partial\Omega_I) = \partial\Omega_P$. That is, we allow any mapping ϕ that maps the boundary of the initial domain onto the boundary of the physical domain. Recall that in the standard approach the mapping ϕ is

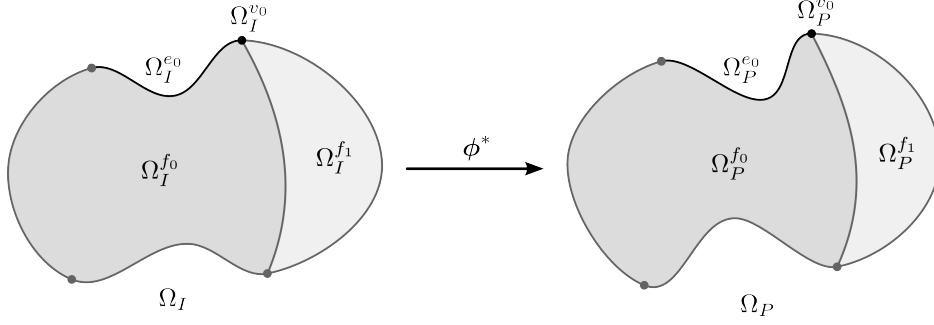


Figure 2: Initial domain Ω_I , composed of four vertices, five curves and two surfaces, mapped to its physical counterpart, Ω_P .

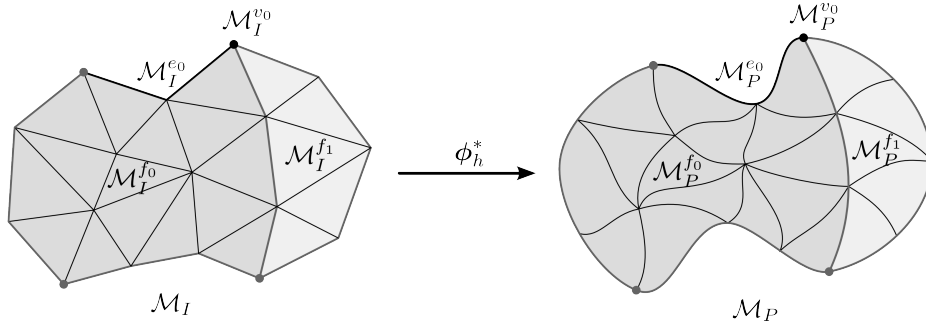


Figure 3: Initial mesh \mathcal{M}_I , mapped to its physical counterpart, \mathcal{M}_P , where each sub-entity of the initial and final domain are approximated using a mesh.

known at the domain boundary which, is a particular case of the mappings we allow. Accordingly, our method can explore more valid diffeomorphisms and therefore, can obtain volume meshes with a smaller distortion.

We assume that both the initial domain and the physical domain, Ω_I and Ω_P , respectively, are composed of the union of a set of sub-entities (namely, the vertices, curves, surfaces and volumes):

$$\Omega_I = \bigcup_{k=1}^N \Omega_I^k, \quad \Omega_P = \bigcup_{k=1}^N \Omega_P^k,$$

where N is the number of sub-entities contained in each domain. Figure 2 shows an initial domain, Ω_I , composed of four vertices, five curves and two surfaces, and a mapping ϕ^* that converts the initial domain into the physical domain, Ω_P . Note that, in addition, each sub-entity of the initial domain, Ω_I^k , has to be mapped into the corresponding sub-entity of the physical domain, Ω_P^k . That is, $\phi^*(\Omega_I^k) = \Omega_P^k$, for $k = 1, \dots, N$. The main idea is that the mapping ϕ has to convert the vertices, edges, faces and volumes of the initial domain into the corresponding counterpart in the physical domain.

The continuous problem consists on finding a diffeomorphism $\phi^* \in \mathcal{C}^1(\Omega_I, \Omega_P)$ such that the distortion measure is ideal and it maps each sub-entity of the initial domain, Ω_I^k , into the corresponding sub-entity of the physical domain, Ω_P^k . Specifically,

$$\begin{aligned} M\phi^* &= 1, & \forall \mathbf{y} \in \Omega_I, \\ \phi^*(\Omega_I^k) &= \Omega_P^k, & \forall \Omega_I^k \in \Omega_I, \end{aligned} \quad (5)$$

5.3. Discrete framework

We consider that the domain Ω_I is approximated by a mesh \mathcal{M}_I in such a way that each sub-entity of the initial domain, Ω_I^k , is approximated by a sub-mesh of \mathcal{M}_I , denoted as \mathcal{M}_I^k . That is, each geometrical entity is discretized using the corresponding mesh, see Figure 3. Note that this is the case of a standard hierarchical mesh generator. The three-dimensional mesh approximates the domain volumes, and the sub-meshes approximate the curves and surfaces that compose the CAD model. In this setting, we seek for an optimal element-wise local diffeomorphism, $\phi_h^* \in \mathcal{C}^0(\Omega_I, \Omega_P)$, such that the physical mesh \mathcal{M}_P has an ideal distortion measure. In addition, we also require that the set of mesh nodes of each sub-entity in the ideal configuration,

$\mathcal{V}(\mathcal{M}_I^k)$, are mapped into the corresponding sub-entity of the physical domain. That is, we impose that the nodes of the physical mesh interpolate the sub-entities of the physical domain. Specifically,

$$M\phi_h^* = 1, \quad \forall \mathbf{y} \in \mathcal{M}_I, \quad (6)$$

$$\mathcal{V}(\mathcal{M}_P^k) = \phi_h^*(\mathcal{V}(\mathcal{M}_I^k)) \in \Omega_P^k, \quad \forall \mathcal{M}_I^k \in \mathcal{M}_I. \quad (7)$$

Since we are looking for a conformal mesh, we seek the mapping ϕ_h^* in the space of vector functions

$$\mathcal{U} = \left\{ \mathbf{u} \in [\mathcal{C}^0(\mathcal{M}_I)]^d \text{ such that } \mathbf{u}|_{e_I} \in [\mathcal{P}^p(e_I)]^d, \forall e_I \in \mathcal{M}_I \right\},$$

where $\mathcal{P}^p(e_I)$ is the space of polynomials on e_I of degree at most p . This corresponds to the standard function space for continuous Galerkin methods, determined by element-wise polynomials of degree at most p and continuous at the element interfaces. Note that for a given initial mesh, \mathcal{M}_I , and a configuration of the physical domain, Ω_P , it may not exist a mapping ϕ_h^* such that Equations (6) and (7) are verified. For this reason, we impose the optimality condition in a least-squares sense:

$$\phi_h^* = \underset{\phi_h \in \mathcal{U}_D}{\operatorname{argmin}} \|M\phi_h - 1\|_{\mathcal{M}_I}^2, \quad (8)$$

where

$$\mathcal{U}_D = \left\{ \phi_h \in \mathcal{U} \text{ such that } (M\phi_h - 1) \in \mathcal{L}^2(\Omega_I), \text{ and } \phi_h(\mathcal{V}(\mathcal{M}_I^k)) \in \Omega_P^k, \forall \mathcal{M}_I^k \in \mathcal{M}_I \right\}.$$

That is, we seek the mapping ϕ_h^* in the set of functions that have a finite distortion measure in the sense of the \mathcal{L}^2 norm (it is a valid mapping) and, in addition, map the nodes of each sub-mesh of the initial domain into the corresponding entity of the physical domain (the nodes interpolate the boundary).

In Equation (8), we define the norms:

$$\|f\|_{\mathcal{M}_I} = \sqrt{\langle f, f \rangle_{\mathcal{M}_I}}, \quad (9)$$

$$\|f\|_{e_I} = \sqrt{\langle f, f \rangle_{e_I}}, \quad (10)$$

being the inner product of scalar functions f and g in \mathcal{M}_I

$$\langle f, g \rangle_{\mathcal{M}_I} = \sum_{e_I \in \mathcal{M}_I} \langle f, g \rangle_{e_I},$$

$$\langle f, g \rangle_{e_I} = \int_{e_I} f(\mathbf{y}) g(\mathbf{y}) d\mathbf{y}.$$

This is the classical inner product of functions in the $\mathcal{L}_2(\Omega_I)$ function space, and it induces the norm of functions introduced in Equation (9). When the optimal mapping ϕ_h^* is found, the physical mesh is characterized as $\mathcal{M}_P = \phi_h^*(\mathcal{M}_I)$. In addition, the meshes that approximate the different sub-entities of the physical domain are defined as $\mathcal{M}_P^k = \phi_h^*(\mathcal{M}_I^k)$, for $k = 1, \dots, N$. Note that in contrast with the classical approaches, the boundary mesh is not fixed, and is the mapping ϕ_h^* who determines the boundary mesh. Specifically, the boundary nodes are free to slide on the corresponding domain sub-entities.

The distortion measure of a given element, e_P , is defined as

$$\eta(e_P) = \frac{\|M\phi_h\|_{e_I}}{\|1\|_{e_I}}, \quad (11)$$

being e_I , the corresponding ideal element. Note that $\|1\|_{e_I}$ is the measure of the ideal element. Then, the quality measure of an element, e_P , can be defined as the inverse of the distortion

$$q(e_P) = \frac{1}{\eta(e_P)}. \quad (12)$$

6. Hierarchical high-order mesh optimization

6.1. Base objective function

The minimization problem stated in (8) can be expressed in terms of elemental contributions, by using Equations (9) and (10), in the following manner:

$$\|M\phi_h - 1\|_{\mathcal{M}_I}^2 = \sum_{e_I \in \mathcal{M}_I} \|M\phi_h - 1\|_{e_I}^2.$$

In the case of nodal high-order elements, the mapping ϕ_h depends on the physical coordinates of the nodes. Thus, the distortion of a mapping ϕ_h can be written in terms of the physical coordinates of the nodes

$$f(\mathbf{x}_1, \dots, \mathbf{x}_N) = \|M\phi_h(\mathbf{y}; \mathbf{x}_1, \dots, \mathbf{x}_{n_t}) - 1\|_{\mathcal{M}_I}^2, \quad (13)$$

where n_t is the total number of nodes of the high-order mesh. For this reason, the computation of ϕ_h^* in Equation (8) is equivalent to determine the position of the physical nodes that minimize the distortion of the mapping ϕ_h . In our setup, we assume that the mapping ϕ_h^* has the form:

$$\phi_h^*(\mathbf{y}) = \sum_{v \in \mathcal{V}(\mathcal{M}_P)} \mathbf{x}_v N_v(\mathbf{y}),$$

where \mathbf{x}_v are the physical coordinates of node v , and $\{N_v\}_{v \in \mathcal{V}(\mathcal{M}_P)}$ is a Lagrange polynomial base of shape functions.

Note that the nodes of the physical mesh can be divided in the set of free nodes that are able to be moved in order to optimize the objective function, and the set of fixed nodes that are not able to be moved. Thus, we can write the function

$$f(\mathbf{x}_1, \dots, \mathbf{x}_{n_f}; \mathbf{p}_1, \dots, \mathbf{p}_{n_p}) = \frac{1}{2} \|M\phi_h - 1\|_{\mathcal{M}_I}^2, \quad (14)$$

where n_f is the number of free nodes, and n_p is the number of prescribed nodes, and $n_f + n_p = n_t$. In our case, the only fixed nodes are the ones that belong to the vertices of the geometry.

According to Equation (7), we need to impose that the boundary nodes in the physical mesh belong to the corresponding geometrical entity of the physical domain. To avoid non-robust and expensive projections from the computational point of view, we describe the objective function (14) in terms of the parametric coordinates of the boundary nodes. That is:

$$\begin{aligned} \mathbf{x}_j &= \varphi_j(\mathbf{u}_j), & j &= 1, \dots, n_s \\ \mathbf{x}_k &= \gamma_k(t_k), & k &= 1, \dots, n_c, \end{aligned}$$

where n_s and n_c are the number of mesh points on the surfaces and curves of the geometry, respectively, \mathbf{u}_j and t_k are the parametric coordinates of surface and curve nodes, respectively, and φ_j and γ_k the parameterizations of the geometric entity where they belong. Using the parametric coordinates of the boundary nodes, the global objective function can be expressed as:

$$f(\mathbf{x}_1, \dots, \mathbf{x}_{n_i}, \mathbf{u}_1, \dots, \mathbf{u}_{n_s}, t_1, \dots, t_{n_c}; \mathbf{p}_1, \dots, \mathbf{p}_{n_p}) = \frac{1}{2} \|M\phi_h - 1\|_{\mathcal{M}_I}^2, \quad (15)$$

where n_i is the number of inner nodes, and $n_i + n_s + n_c = n_f$. Equation (15) describes a global objective function that depends on the position of all the mesh nodes, and it is optimized by moving the location of the inner nodes, the location of the surface nodes and the location of the curve nodes, while keeping fixed the position of vertex nodes.

6.2. Interior nodes objective function

We have deduced a global objective function that depends on the coordinates of all the free mesh nodes. Since we use nodal elements, we can deduce

a local objective function that only depends on the coordinates of one specific node, see details in [22]. Thus, the local objective function allows the implementation of a non-linear Gauss-Seidel procedure. The local objective function for a single node, v , is defined as the addition of the elemental contributions of the adjacent elements:

$$f_v = \sum_{e_I \in \mathcal{M}_v} \frac{1}{2} \|M\phi_h - 1\|_{e_I}^2, \quad (16)$$

where \mathcal{M}_v is the set of elements adjacent to node v , and $\phi_{e_I}^*$ is the mapping between the ideal element, e_I , and the physical element, e .

6.3. Curve nodes objective function

We need to impose that the curve nodes in the smoothed mesh are located on the boundary curves. To this end, we rewrite the objective function (16) in terms of the parametric coordinates of the boundary nodes. Given a parametric curve, $\gamma(t)$, and a node on the curve, v , the corresponding objective function is expressed as the composition of functions:

$$f_\gamma(t) = f_v(\gamma(t)) \quad (17)$$

The derivatives of the objective function (17) with respect to the parametric coordinate t can be expressed in terms of Function (16) and the curve parameterization, $\gamma(t)$, by means of the chain rule. That is,

$$\nabla f_\gamma(t) = \nabla f_v(\gamma(t)) \cdot \gamma'(t), \quad (18)$$

and

$$Hf_\gamma(t) = (\gamma')^T \cdot \nabla f_v(\gamma(t)) \cdot \gamma' + \nabla f_v(\gamma(t)) \cdot \gamma''. \quad (19)$$

The derivatives of the curve parameterization, γ , are provided by the CAD engine. In our applications we use OpenCascade as the geometric engine [51]. Note that we only need to compute the derivatives of the objective function (16) to restrict the movement of a node on a curve, since the derivatives of the curve parameterization are provided by the CAD engine. For this reason, given a generic objective function, it is straightforward to obtain a new one for a node that belongs to a curve.

6.4. Surface nodes objective function

Given a node on a surface, v , we define the corresponding objective function by composing (16) with the surface parameterization, $\varphi(\mathbf{u})$. That is:

$$f_\varphi(\mathbf{u}) = f_v(\varphi(\mathbf{u})). \quad (20)$$

The derivatives of the objective function (20) with respect to coordinates \mathbf{u} are deduced by means of the chain rule. That is,

$$\nabla f_{\varphi}(\mathbf{u}) = \mathbf{D}\varphi(\mathbf{u}) \cdot \nabla f_v(\varphi(\mathbf{u})), \quad (21)$$

and

$$Hf_{\varphi}(\mathbf{u}) = H\varphi(\mathbf{u}) : \nabla f_v(\varphi(\mathbf{u})) + (\mathbf{D}\varphi(\mathbf{u}))^T \cdot \nabla f_v(\varphi(\mathbf{u})) \cdot \mathbf{D}\varphi(\mathbf{u}), \quad (22)$$

where $H\varphi(\mathbf{u})$ is a third order tensor whose components are defined as

$$[H\varphi(\mathbf{u})]_{ijk} = \frac{\partial^2 \varphi_k(\mathbf{u})}{\partial u_i \partial u_j},$$

and

$$[H\varphi(\mathbf{u}) : \nabla f_v(\varphi(\mathbf{u}))]_{ij} = \sum_{k=1}^n \frac{\partial f(\varphi(\mathbf{u}))}{\partial x_k} \cdot \frac{\partial^2 \varphi_k(\mathbf{u})}{\partial u_i \partial u_j}.$$

Similar to the curve nodes case, we only need to compute the derivatives of the original objective function, because the derivatives of the surface parameterization are provided by the CAD engine.

6.5. Optimization approach

In this section, we detail the optimization approach to obtain the new position of the nodes that untangles the mesh and improves the element quality. It is important to highlight that only a single distortion measure, the mesh volume distortion, is used in the whole process. The method is devised as an iterative process that minimizes the objective function (15), moving one node at a time, see Algorithm 1. The nodes are divided in three sets, according to their location. First, we smooth the curve nodes, prescribing the position of all the other nodes, Lines 5–9. Second, we smooth the surface nodes by assuming that the positions of the remaining nodes are fixed, Lines 10–14. Third, we smooth the interior nodes fixing the position of all the other nodes, Lines 15–19. For each node, v , the appropriate objective function is used, depending on its location. The process is iterated until the maximum displacement of the nodes is below a threshold prescribed by the user, Line 20. Accordingly, the number of times the optimization process cycles through curves, surfaces, and volumes, depends on the prescribed maximum displacement. That is, the nodes on each curve, surface and volume are

Algorithm 1 Procedure to smooth a mesh, \mathcal{M}_P .

```

1: function smoothMesh(Mesh  $\mathcal{M}_P$ , Real  $\varepsilon$ )
2:   Boolean isConverged  $\leftarrow$  false
3:   while not isConverged do
4:     Real disp  $\leftarrow$  0
5:     for all  $v \in \mathcal{E}$  do ▷ Edge nodes
6:       Function  $f_{\gamma} \leftarrow$  getCurveFunction ▷ Eq. (17)
7:       Real nodeDisp  $\leftarrow$  smoothNode( $v, f_{\gamma}$ )
8:       disp  $\leftarrow$  max{disp, nodeDisp}
9:     end for
10:    for all  $v \in \mathcal{F}$  do ▷ Face nodes
11:      Function  $f_{\varphi} \leftarrow$  getSurfaceFunction ▷ Eq. (20)
12:      Real nodeDisp  $\leftarrow$  smoothNode( $v, f_{\varphi}$ )
13:      disp  $\leftarrow$  max{disp, nodeDisp}
14:    end for
15:    for all  $v \in \mathcal{V}$  do ▷ Volume nodes
16:      Function  $f_v \leftarrow$  getBaseFunction ▷ Eq. (16)
17:      Real nodeDisp  $\leftarrow$  smoothNode( $v, f_v$ )
18:      disp  $\leftarrow$  max{disp, nodeDisp}
19:    end for
20:    isConverged  $\leftarrow$  (disp  $\leq$   $\varepsilon$ )
21:  end while
22: end function

```

relocated several times until global convergence is achieved.

Note that in Algorithm 1 we update the location of one node at a time, and it is computed taking into account the new location of the previously relocated nodes. In addition, nodes are processed according to the entity they belong to: first, we move the curve nodes; second, the surface nodes; and third, the volume nodes. Thus, the proposed implementation of the minimization process can be understood as a non-linear Gauss-Seidel procedure by blocks (curves, surfaces, and volumes).

The position of a node is computed using a line-search iterative process. Let $\mathbf{x}_{\Omega_i}^k$ be the position of node v that belongs to domain Ω_i at iteration k . Note that $\mathbf{x}_{\Omega_i}^k$ may refer to the position of a node in the physical space, \mathbf{x}_v , the parametric coordinates of a node in a surface \mathbf{u}_v , or the parametric coordinates of a node in a curve, t_v . The new position at iteration $k + 1$ is defined as

$$\mathbf{x}_{\Omega_i}^{k+1} = \mathbf{x}_{\Omega_i}^k + \alpha^k \mathbf{p}^k, \quad (23)$$

where \mathbf{p}^k is an advancing direction and α^k is a

Algorithm 2 Backtracking line-search procedure.

```
1: function LineSearch(Point  $\mathbf{x}_{\Omega_i}^k$ , Function  $f_{\Omega_i}$ )
2:   Matrix  $H \leftarrow H f_{\Omega_i}(\mathbf{x}_{\Omega_i}^k)$ 
3:   Vector  $\mathbf{p}^k \leftarrow -H^{-1} \nabla f_{\Omega_i}(\mathbf{x}_{\Omega_i}^k)$ 
4:   Real  $\alpha \leftarrow 1$ , Real  $\rho \leftarrow 0.5$ , Real  $c \leftarrow 10^{-4}$ 
5:   while  $f_{\Omega_i}(\mathbf{x}_{\Omega_i}^k + \alpha \mathbf{p}^k) \geq f_{\Omega_i}(\mathbf{x}_{\Omega_i}^k) +$   

    $c \alpha \mathbf{p}^k \nabla f_{\Omega_i}(\mathbf{x}_{\Omega_i}^k)$  do
6:      $\alpha \leftarrow \rho \alpha$ 
7:   end while
8:    $\mathbf{x}_{\Omega_i}^{k+1} \leftarrow \mathbf{x}_{\Omega_i}^k + \alpha \mathbf{p}^k$ 
9:   return  $\mathbf{x}_{\Omega_i}^{k+1}$ 
10: end function
```

step length. To compute both the advancing direction and the step length, there are several methods, like the steepest descent or the Newton-Raphson methods, see [55]. Algorithm 2 details our particular implementation of the backtracking line-search method. The input of the method are the coordinates of a node v in a domain Ω_i at iteration k , $\mathbf{x}_{\Omega_i}^k$, and the corresponding objective function to minimize for the node, f_{Ω_i} . Note that f_{Ω_i} has to match the type of the current node. Therefore, we use f_γ for curve nodes, f_φ for surface nodes, and f_v for the inner nodes. The line-search method is performed in two steps. First, we compute an advancing direction using the Newton-Raphson method, Lines 2 and 3. Note that to apply this algorithm, the first and the second derivatives of the objective function may be needed, and they are detailed in Appendix A. Second, to compute the step length, we use the Wolfe conditions to ensure that a sufficient decrease in the objective function is achieved, Lines 4–8. The parameter values of the backtracking method, Line 4, are defined as detailed in Reference [55].

7. Examples

In this section, we present several examples that illustrate the advantages of the proposed method to untangle and smooth high-order tetrahedral meshes. For each example we show the CAD model, the initial mesh, the inverted elements of the initial mesh and the smoothed mesh. In addition, for each example we also present a detailed view of the initial linear mesh and the final curved high-order mesh. Finally, we present a table that summarizes the shape quality, see Equation (12), statistics of the mesh elements. Specifically, we provide

the number of tangled elements, and the minimum, the maximum, the mean and the standard deviation of the element quality. We highlight that in all cases the smoothed mesh increases the minimum and mean values of the mesh quality and decreases its standard deviation. Note that the proposed objective function corresponds to an \mathcal{L}^2 norm and therefore, it improves the average element distortion. The objective function penalizes, by construction, the highly distorted elements. This is in agreement with the obtained quality statistics, since the average and minimum quality increase. In addition, note that the maximum quality may decrease to accommodate the improvement of the whole mesh.

7.1. High-order mesh for a ratchet gear ($p = 6$)

This example presents a tetrahedral mesh of polynomial degree six for a ratchet gear, shown in Figure 4(a). Note that this geometry contains high-curvature surfaces that compromise the generation of a valid high-order mesh. Specifically, there are several fillets and, in addition, the inner holes and the central hole contain curved surfaces. The initial curved high-order mesh is shown in Figure 4(b). This mesh is not valid since it contains a large number of inverted elements around the curved surfaces, see Figure 4(c). When we apply the proposed hierarchical smoother, we obtain a high-quality mesh without inverted elements, see Figure 4(d). To highlight the benefits of the high-order discretization, Figure 5(a) shows a detail of the initial linear mesh and Figure 5(b) shows the corresponding high-order curved mesh. Note that the latter better approximates the curved curves and surfaces of the geometry, because of the higher polynomial degree used to interpolate the geometry.

Table 1 shows the element quality statistics of the initial and smoothed high-order meshes. The high-order mesh contains 10 324 elements, and 825 of them are inverted. Thus, the minimum element quality of the mesh is 0. When we apply the hierarchical smoother, the final mesh does not contain any inverted element. The minimum quality is 0.40, and both the maximum and mean quality are increased while the standard deviation is decreased. Thus, the smoothed high-order mesh contains better quality elements than the original mesh.

7.2. High-order mesh for a break disk ($p = 5$)

In this example, we generate a high-order mesh of polynomial degree five for a break disk, shown

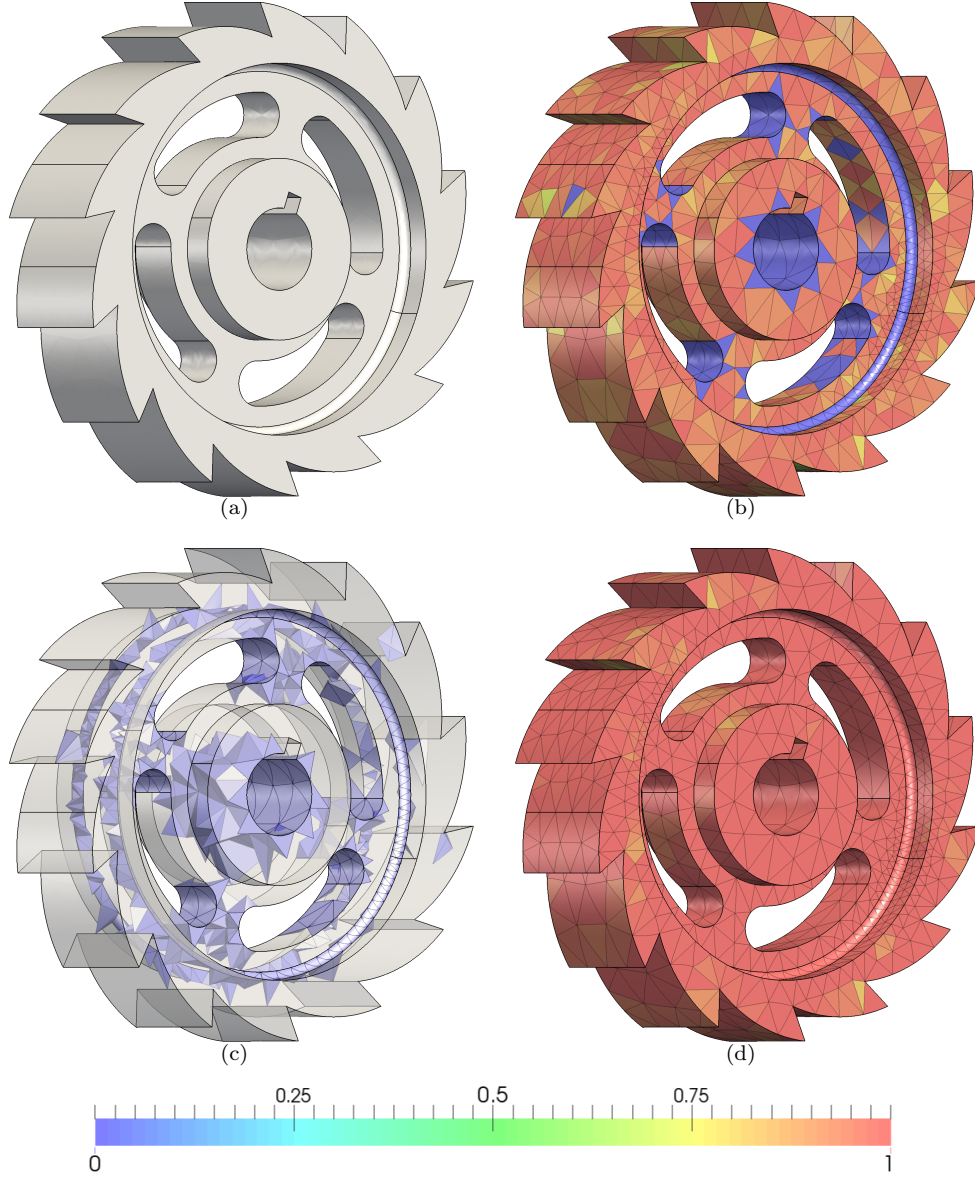


Figure 4: High-order tetrahedral mesh ($p = 6$) for the ratchet gear: (a) geometry definition (CAD); (b) initial curved high-order mesh; (c) inverted elements in the initial curved high-order mesh; and (d) high-order mesh after applying the hierarchical smoothing.

in Figure 6(a). This geometry is very thin and, for this reason, the smoothing process of a high-order mesh may fail because the feasible region of the nodes is small. Note that this mesh contains between one and two layers of elements almost everywhere, but around the high-curvature surfaces that define the holes. If a classical approach is applied, where first each surface is smoothed separately, since the opposing surfaces are very close,

it may not be possible to generate a valid mesh. The proposed hierarchical approach deals with this kind of situations by taking into account only the quality of the volume mesh, even for the nodes that are on the boundary. The initial curved high-order mesh contains inverted elements and, for this reason, it is not valid for simulation purposes, see Figure 6(b). The inverted elements are located around the surfaces of the domain, see Figure 6(c). After

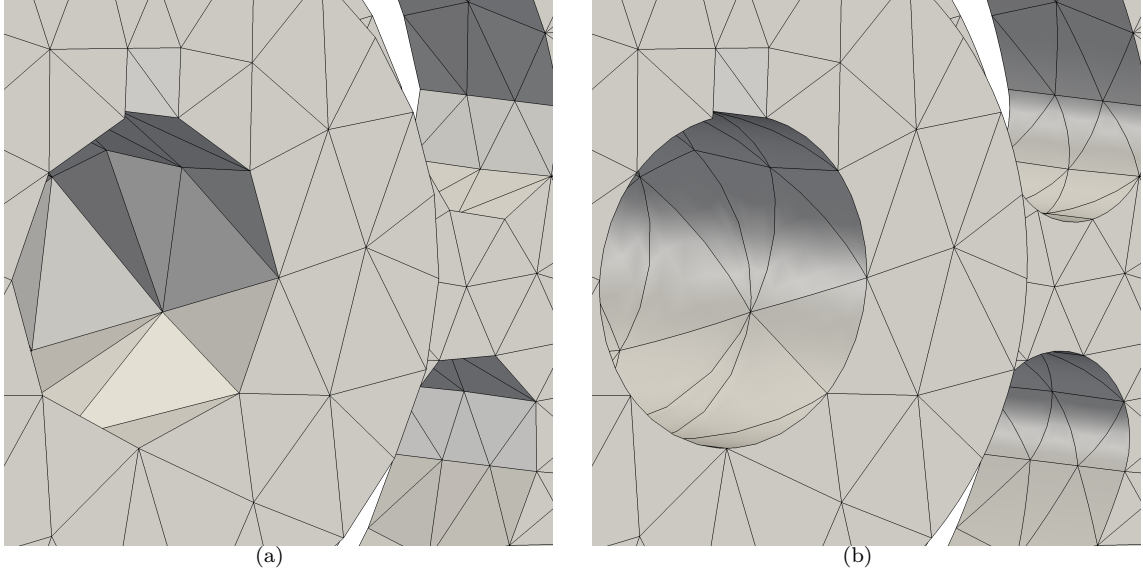


Figure 5: Comparison of the initial linear mesh and the final high-order curved mesh for the ratchet gear: (a) initial linear mesh; and (b) final curved high-order mesh of polynomial degree six.

applying the proposed hierarchical smoothing, we obtain a high-order mesh without any inverted element, see Figure 6(d). In this example we have also applied the standard smoothing approach in which first the boundary nodes are moved by taking into account the quality of the boundary triangles and then the inner nodes are moved taking into account the quality of the tetrahedral elements. Note that in this case, a valid high-order mesh has not been obtained because the boundary elements constrain the quality of the whole mesh. Figures 7(a) and 7(b) show the inverted elements and the element quality after applying the standard smoothing approach, respectively. In addition, Figures 8(a) and 8(b) show a detailed view of the initial linear mesh and the final high-order curved mesh for the propeller.

Table 1 presents the statistical information of the element quality for the meshes of this example. The original mesh is composed of 40 330 elements and contains 1 313 inverted elements thus, the minimum quality is 0. After applying the hierarchical smoother, the high-order mesh does not contain any inverted elements. The minimum quality of the smoothed mesh has been increased to 0.50. In addition, the maximum quality have almost retained the same value and the mean quality has been increased from 0.89 to 0.95. Note that the mesh obtained after applying the standard smoothing approach con-

tains 10 508 inverted elements. Thus, the minimum quality is 0 and a invalid mesh is obtained

7.3. High-order mesh for a propeller ($p = 3$)

The last example shows the mesh generated of polynomial degree three for both the exterior domain and the interior domain of a five-bladed propeller, see Figure 9(a). The initial curved high-order mesh, see Figure 9(b) contains inverted elements in both volumes, see Figure 9(c). In this example, instead of applying the hierarchical smoother to each volume separately, we apply it to the global mesh, composed of the mesh for each volume. In this way, the objective function of the nodes that belong to the interface takes into account the quality of all surrounding elements, even when they belong to different volume meshes. The smoothed mesh, shown in Figure 9(d) does not contain any inverted element. Figures 10(a) and 10(b) compare the initial linear and high-order curved meshes generated for the propeller. Note that the high-order curved mesh better reproduces the geometry because of the higher polynomial degree of the shape functions.

Table 1 shows the statistical information for the meshes presented in this example. The initial mesh contains 338 192 elements, both in the inner and the outer part of the geometry. In addition, 266

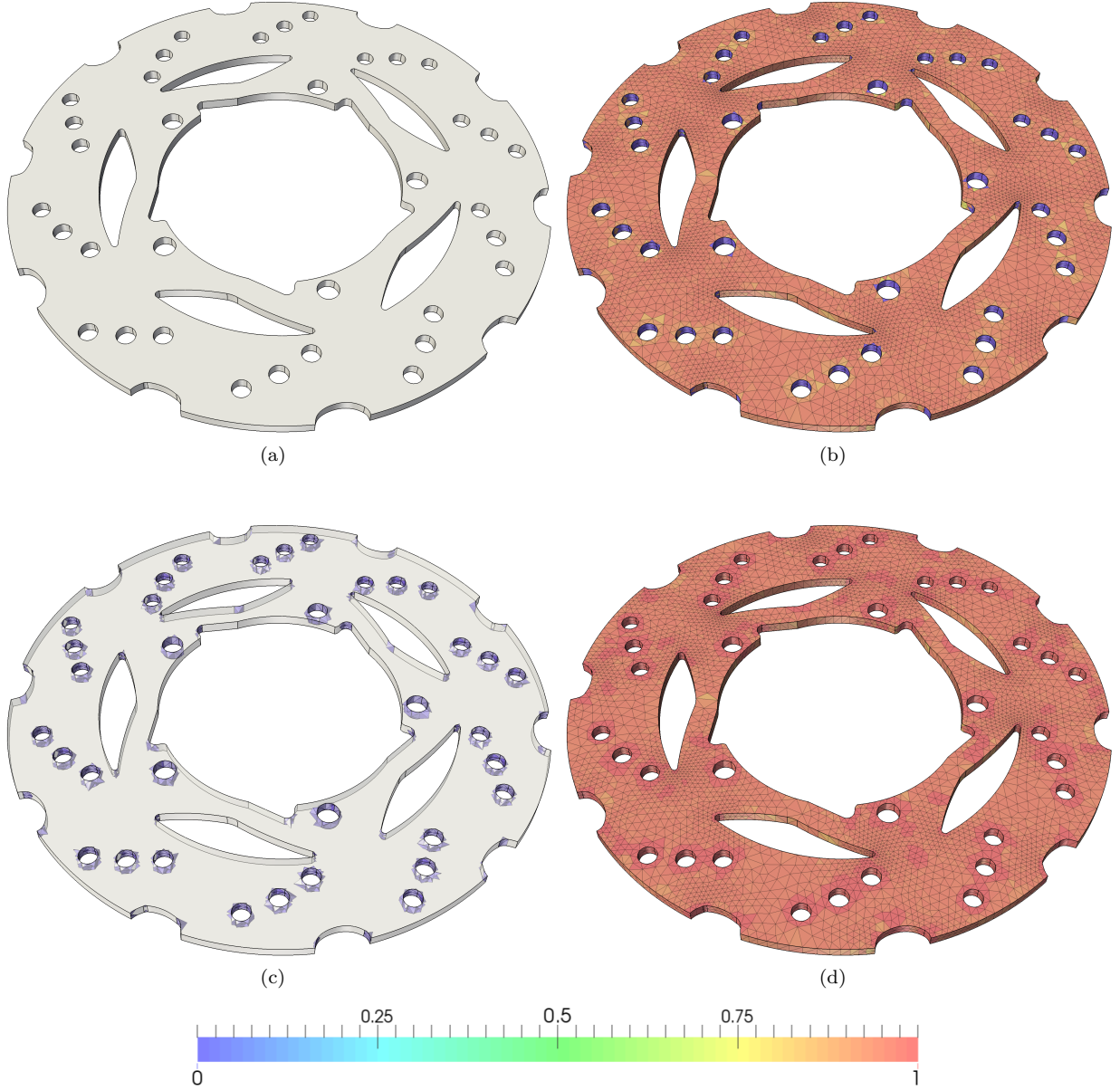


Figure 6: High-order tetrahedral mesh ($p = 5$) for the break disk: (a) geometry definition (CAD); (b) initial curved high-order mesh; (c) inverted elements in the initial curved high-order mesh; and (d) high-order mesh after applying the hierarchical smoothing.

elements are inverted and, for this reason, the minimum quality is 0. When the hierarchical smoother is applied, we obtain a high-quality mesh without inverted elements, in which the minimum quality is 0.18. In this case, the maximum quality, the mean quality and the standard deviation are about the same for the initial and the smoothed meshes. Note

however, that the initial mesh cannot be used for simulation purposes because it contains inverted elements.

8. Conclusions

We have presented a novel untangling and smoothing method to curve high-order meshes. The

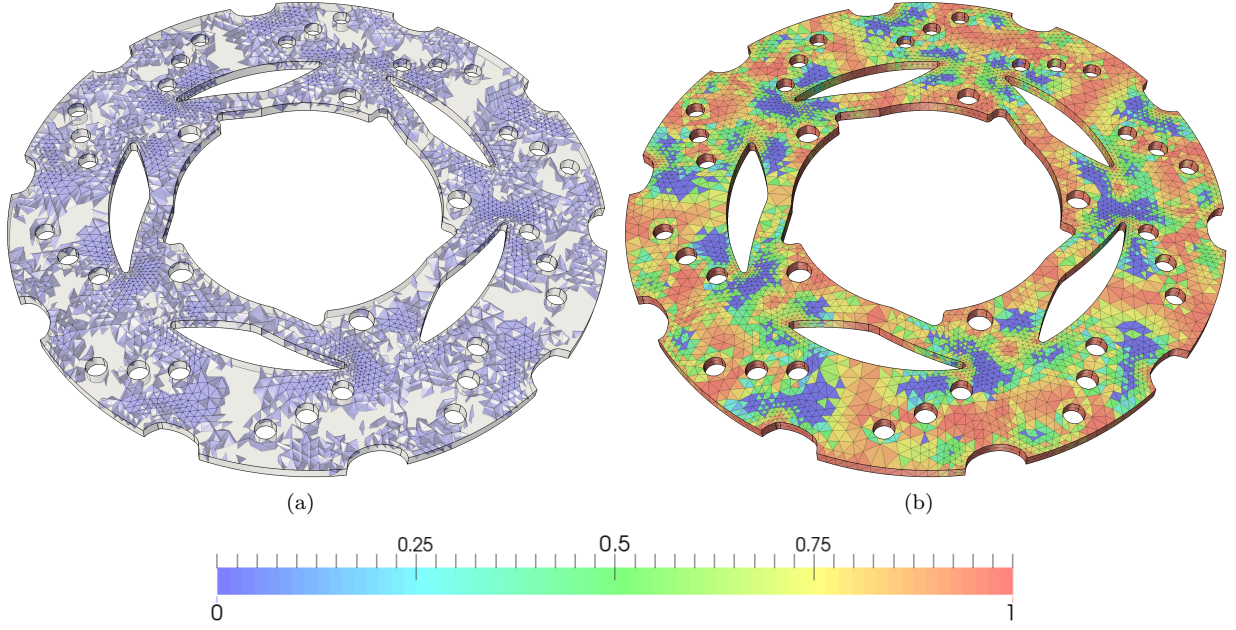


Figure 7: High-order tetrahedral mesh ($p = 5$) for the break disk smoothed using the standard approach: (a) inverted elements; and (b) element quality.

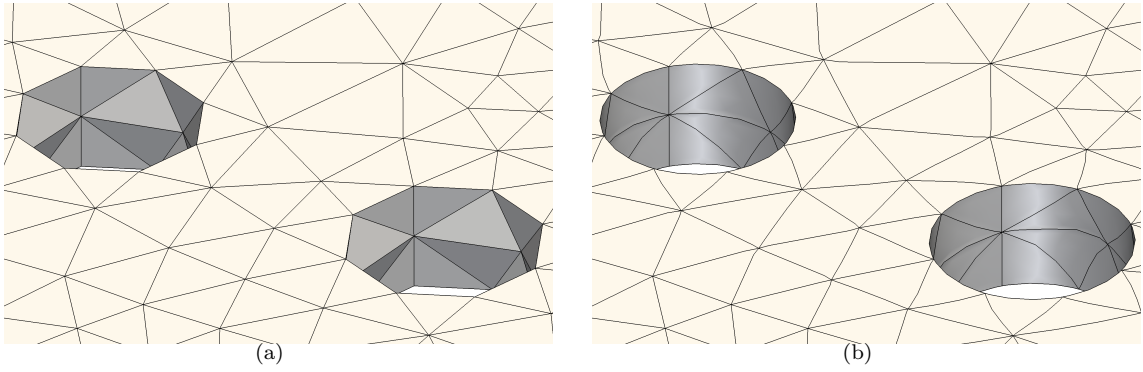


Figure 8: Comparison of the initial linear mesh and the final high-order curved mesh for the break disk: (a) initial linear mesh; and (b) final curved high-order mesh of polynomial degree five.

Table 1: Element quality statistics of the presented high-order meshes.

	Ratchet gear ($p = 6$, 10 324 elems)		Break disk ($p = 5$, 40 330 elems)			Propeller mesh ($p = 3$, 338 192 elems)	
	initial	smoothed	initial	smoothed	standard	initial	smoothed
inverted elem.	825	0	1 313	0	10 508	266	0
minimum	0	0.40	0	0.50	0	0	0.18
maximum	0.92	0.99	0.96	0.99	0.99	0.94	0.99
mean	0.75	0.97	0.89	0.95	0.66	0.90	0.93
deviation	0.24	0.05	0.17	0.04	0.42	0.06	0.05

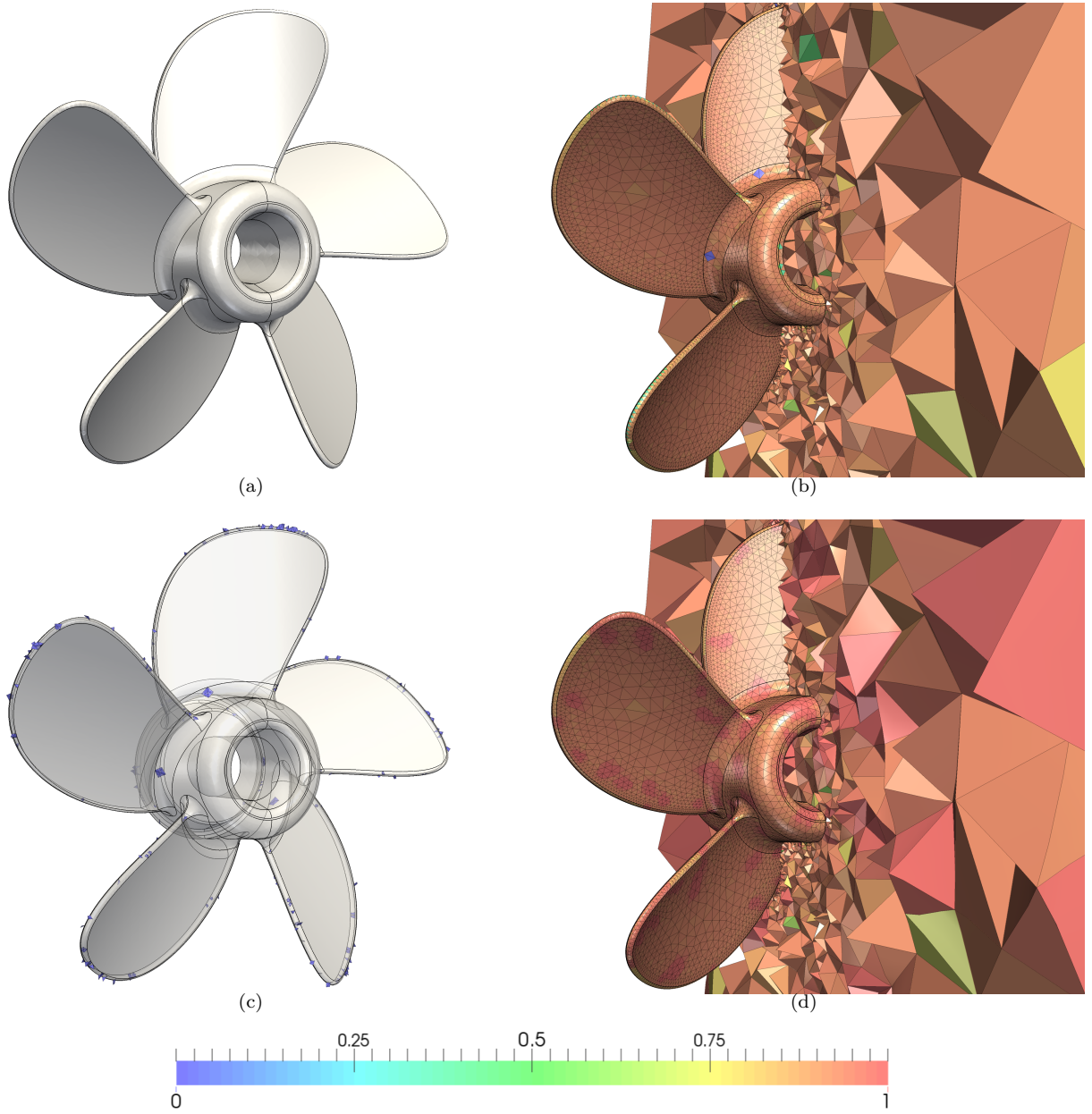


Figure 9: High-order tetrahedral mesh ($p = 4$) for the propeller: (a) geometry definition (CAD); (b) initial curved high-order mesh; (c) inverted elements in the initial curved high-order mesh; and (d) high-order mesh after applying the hierarchical smoothing.

untangling and smoothing process is accomplished by optimizing a single objective function defined in terms of a regularized distortion measure of the volume mesh. The main contribution is that we use a single target objective function in the whole process. On the contrary, the standard mesh curving approaches apply two separate curving methods

with their corresponding associated residual systems. The first stage takes into account a curving criteria for surface meshes, while the second one takes into account a curving criteria for volume meshes. In our approach, the target of the proposed objective function is the distortion of the volume mesh. Thus, we are able to obtain meshes com-

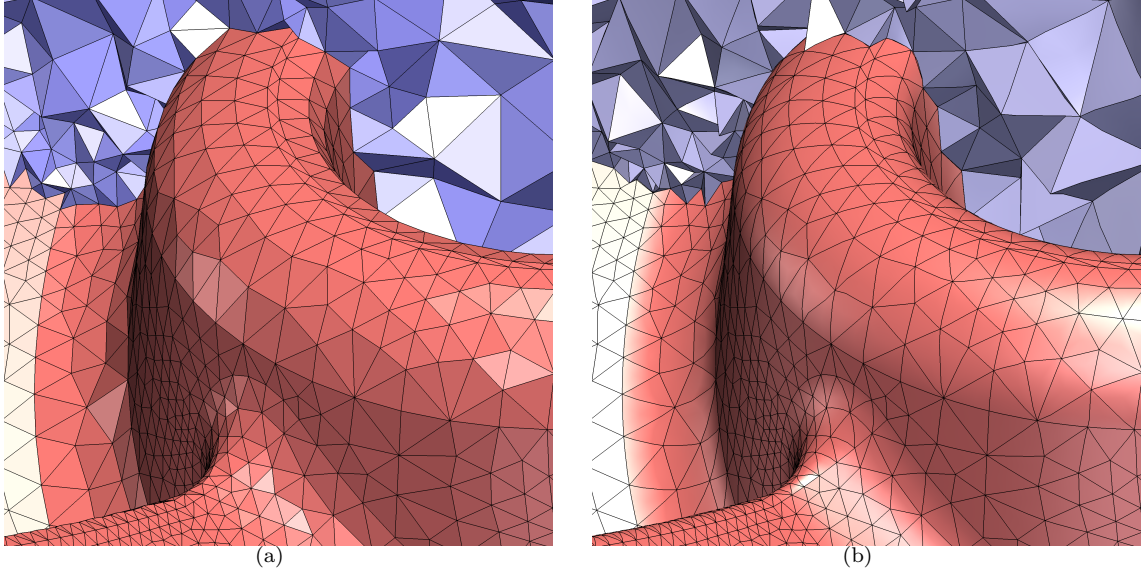


Figure 10: Comparison of the initial linear mesh and the final high-order curved mesh for the propeller: (a) initial linear mesh; and (b) final curved high-order mesh of polynomial degree three.

posed of better quality elements, specially when the boundary mesh constrains the quality of the whole mesh, like in thin regions of the geometry. However, in order to obtain a mesh without inverted elements, we need to move the interior nodes of the mesh, as well as the boundary nodes. With this objective in mind, we have deduced an objective function that takes the physical coordinates of the inner nodes and the parametric coordinates of the boundary nodes as input parameters. Finally, it is worth to notice that the proposed hierarchical approach can be directly applied to meshes of assembly models, composed of several volumes. Accordingly, the objective function for the nodes on the interface between volumes takes into account the quality of the all the surrounding elements, even when they are located in different volumes.

Although we have deduced a global objective function, for implementation purposes we have adopted a local optimization approach. That is, instead of moving all the nodes at the same time, we perform a node-by-node iteration until convergence is achieved. This allows to perform a hierarchical smoothing approach which is divided into three stages. First, we move the nodes on the curves; second, we move the nodes on the surfaces; and third, we move the inner nodes. Since we did not prove that the proposed objective function is convex, there is no convergence guarantee. In practical

applications, we use backtracking line search to improve the global convergence features of the implementation. It is important to point out that with all the tested examples we have converged to a local minima that provides a valid mesh, and we did not experience non-convergent or oscillatory behavior during the optimization process.

The proposed method is in general terms more expensive than standard smoothing approaches (without untangling) implemented in a hierarchical manner. For instance, the proposed implementation of our method is more expensive than standard node-by-node Laplacian smoothing. That is, the cost of moving a node is also proportional to the number of neighbors, but there are more floating operations involved. Nevertheless, our implementation scales as the node-by-node Laplacian method up to an implementation constant. Note that our formulation may need several smoothing cycles to achieve the minimum of the objective function. On the contrary, standard hierarchical approaches only require one smoothing cycle: first, curve meshes are smoothed while the limiting vertex nodes are fixed; second, surface meshes are smoothed while the boundary curve nodes are fixed; finally, the volume mesh is smoothed while the boundary nodes are fixed. Nevertheless, we highlight that the overhead of our method pays off in those applications where a valid mesh cannot be obtained with a stan-

standard hierarchical smoothing method.

We consider that our method is a relevant alternative over the standard hierarchical approach to curve high-order meshes. Specifically, by smoothing and untangling simultaneously boundary and inner nodes we can generate curved high-order meshes in geometries where other methods fail. Specially, in geometries that feature thin regions or high-curvature values where fixing the boundary nodes may add artificial constraints that impede obtaining a valid mesh. These concerns are properly corroborated by the examples presented in this work.

We have restricted ourselves to geometries delimited by parameterized curves and surfaces, since CAD models are the preferred representation for industrial applications. Accordingly, the algorithm in its current form requires the utilization of a CAD engine that provides the evaluation of the parameterizations and their first and second derivatives. However, there are several applications where the geometry is enclosed by a non-parameterized representation such as a triangular mesh. To extend our method to deal with these non-parameterized representations of the geometry, it would be required to obtain a local parameterization that provides the required derivatives for the optimization method of choice. We point out that this extension would not be straight forward since obtaining such parameterizations is still an open problem and an active area of research.

Several aspects of the algorithm can be extended in the near future. The proposed approach is an optimization based method with untangling capabilities constrained to move the nodes on the CAD entities. Hence, it requires more floating point operations per node than other methods, *e.g.* Laplacian smoothing. It would be interesting to explore additional techniques to reduce the computational cost of smoothing a given mesh. For instance, we could use a parallel version of the presented algorithm to reduce the time to smooth a mesh. With this objective in mind, we need to partition properly the mesh nodes to avoid that different processors access and modify the same node position at the same time. In addition, we have chosen to implement the global function in a node-by-node manner to simplify the solver implementation, reduce the memory footprint, and exploit the local behavior of the objective function. However, we need to explore additional minimization approaches in order to investigate the robustness and the performance of different solvers. One of the most interesting ap-

proaches is to consider a global solver, in which all the nodes are moved at the same time. It is important to point out that the global implementation could lead to an ill-conditioned Hessian matrix since the scales of motion in the curves, surfaces, and volumes are different. This could require to scale the optimization variables to improve the conditioning as proposed in [21].

Appendix A. Derivatives of the shape distortion measure

In this appendix, we detail the first and second derivatives of the objective function for a given node, f_v , see Equation (16). We express the derivatives of the objective function in terms of the derivatives of the modified distortion measure, η_δ . That is:

$$\frac{\partial f_v}{\partial x_i} = \sum_{e \in \mathcal{M}_v} \left\langle \frac{\partial \eta_\delta(\mathbf{D}\phi_e^*)}{\partial x_i}, \eta_\delta(\mathbf{D}\phi_e^*) - 1 \right\rangle_{\mathcal{M}_I}, \quad (\text{A.1})$$

and

$$\begin{aligned} \frac{\partial^2 f}{\partial x_i \partial x_j} = & \sum_{e \in \mathcal{M}_v} \left\langle \frac{\partial^2 \eta_\delta(\mathbf{D}\phi_e^*)}{\partial x_i \partial x_j}, \eta_\delta(\mathbf{D}\phi_e^*) - 1 \right\rangle_{\mathcal{M}_I} + \\ & \left\langle \frac{\partial \eta_\delta(\mathbf{D}\phi_e^*)}{\partial x_i}, \frac{\partial \eta_\delta(\mathbf{D}\phi_e^*)}{\partial x_j} \right\rangle_{\mathcal{M}_I}. \quad (\text{A.2}) \end{aligned}$$

We only need to deduce the first and second derivatives of the modified distortion measure, η_δ , see Equation (2). Applying the chain rule, it can be shown that the first derivative is

$$\frac{\partial \eta_\delta}{\partial x_i} = 2\eta_\delta \left[\frac{\left(\frac{\partial \mathbf{D}\phi_e^*}{\partial x_i}, \mathbf{D}\phi_e^* \right)}{(\mathbf{D}\phi_e^*, \mathbf{D}\phi_e^*)} - \frac{\frac{\partial \sigma}{\partial x_i}}{n(2\sigma_\delta - \sigma)} \right], \quad (\text{A.3})$$

and the second derivative is

$$\begin{aligned} \frac{\partial^2 \eta_\delta}{\partial x_i \partial x_j} = & \frac{1}{\eta_\delta} \frac{\partial \eta_\delta}{\partial x_i} \frac{\partial \eta_\delta}{\partial x_j} + 2\eta_\delta \left[\frac{\left(\frac{\partial \mathbf{D}\phi_e^*}{\partial x_i}, \frac{\partial \mathbf{D}\phi_e^*}{\partial x_j} \right)}{(\mathbf{D}\phi_e^*, \mathbf{D}\phi_e^*)} - \right. \\ & \frac{\left(\frac{\partial \mathbf{D}\phi_e^*}{\partial x_i}, \mathbf{D}\phi_e^* \right) \left(\frac{\partial \mathbf{D}\phi_e^*}{\partial x_j}, \mathbf{D}\phi_e^* \right)}{(\mathbf{D}\phi_e^*, \mathbf{D}\phi_e^*)^2} + \\ & \left. \frac{\frac{\partial \sigma}{\partial x_i} \frac{\partial \sigma}{\partial x_j} \frac{\sigma}{n(2\sigma_\delta - \sigma)^3}}{n(2\sigma_\delta - \sigma)^3} \right]. \quad (\text{A.4}) \end{aligned}$$

Acknowledgements

This research was partially supported by CONACYT-SENER (“Fondo Sectorial CONACYT SENER HIDROCARBUROS”, grand contract 163723). The work of the second author was partially supported by the Boeing Co. & US Air Force Office of Scientific Research & European Commission through the Boeing-MIT Alliance & Computational Math Program & Marie Skłodowska-Curie Actions (HiPerMeGaFlows project), respectively.

References

- [1] B A Szabo and I Babuška. *Finite Element Analysis*. John Wiley & Sons New York, 1991.
- [2] C Schwab. *p-and hp-finite element methods: Theory and applications in solid and fluid mechanics*. Clarendon Press Oxford, 1998.
- [3] M O Deville, P F Fischer, and E H Mund. *High-order methods for incompressible fluid flow*, volume 9. Cambridge University Press, 2002.
- [4] J S Hesthaven and T Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Texts in Applied Mathematics. Springer, 2007.
- [5] G Karniadakis and S Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.
- [6] P E Vos, S Sherwin, and R Kirby. From h to p efficiently: implementing finite and spectral/hp element methods to achieve optimal performance for low- and high-order discretisations. *J. Comput. Phys.*, 229(13):5161–5181, 2010.
- [7] R Löhner. Error and work estimates for high-order elements. *Int. J. Numer. Methods Fluids*, 67(12):2184–2188, 2011.
- [8] M Yano et al. *An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes*. PhD thesis, Massachusetts Institute of Technology, 2012.
- [9] R Kirby, S Sherwin, and B Cockburn. To CG or to HDG: a comparative study. *J. Sci. Comput.*, 51(1):183–212, 2012.
- [10] A Huerta, A Angeloski, X Roca, and J Peraire. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *Int. J. Numer. Methods Eng.*, 96:529–560, 2013.
- [11] R Löhner. Improved error and work estimates for high-order elements. *Int. J. Numer. Methods Fluids*, 72:1207–1218, 2013.
- [12] ZJ Wang, K Fidkowski, R Abgrall, F Bassi, D Caraeni, A Cary, H Deconinck, R Hartmann, K Hillewaert, H T Huynh, et al. High-order cfd methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.
- [13] S Dey, M S Shephard, and J E Flaherty. Geometry representation issues associated with p-version finite element computations. *Comput. Meth. Appl. M.*, 150(1–4):39–55, 1997.
- [14] S Dey, R O’Bara, and M S Shephard. Curvilinear mesh generation in 3D. *Comput. Aided Design*, 33:199–209, 2001.
- [15] X Luo, M S Shephard, J F Remacle, R O’Bara, M Beall, B Szabó, and R Actis. P-version mesh generation issues. In *Proc. 11th Int. Meshing Roundtable*, pages 343–354. Citeseer, 2002.
- [16] S Sherwin and J Peiró. Mesh generation in curvilinear domains using high-order elements. *Int. J. Numer. Meth. Eng.*, 53(1):207–223, 2002.
- [17] P O Persson and J Peraire. Curved mesh generation and mesh refinement using lagrangian solid mechanics. In *Proc. 47th AIAA*, 2009.
- [18] X Roca, A Gargallo-Peiró, and J Sarrate. Defining quality measures for high-order planar triangles and curved mesh generation. In *Proc. 20th Int. Meshing Roundtable*, 2012.
- [19] P L George and H Borouchaki. Construction of tetrahedral meshes of degree two. *Int J Numer Meth Eng*, 90(9):1156–1182, 2012.
- [20] Z Xie, R Sevilla, O Hassan, and K Morgan. The generation of arbitrary order curved meshes for 3D finite element analysis. *Comput. Mech.*, 51:361–374, 2012.
- [21] T Toulorge, C Geuzaine, J F Remacle, and J Lambrechts. Robust untangling of curvilinear meshes. *Journal of Computational Physics*, 254:8–26, 2013.
- [22] A Gargallo-Peiró, X Roca, J Peraire, and J Sarrate. Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. *Int. J. Numer. Meth. Engng*, (accepted), 2015.
- [23] E Ruiz-Gironés, X Roca, and J Sarrate. Optimizing Mesh Distortion by Hierarchical Iteration Relocation of the Nodes on the CAD Entities. In *Proc. 23rd Int. Meshing Roundtable*, 2014.
- [24] A Gargallo-Peiró, X Roca, and Sarrate J. A surface mesh smoothing and untangling method independent of the CAD parameterization. *Computational Mechanics*, 53(4):587–609, 2014.
- [25] A Gargallo-Peiró, X Roca, J Peraire, and J Sarrate. Defining quality measures for validation and generation of high-order tetrahedral meshes. In *Proc. 22nd Int. Meshing Roundtable*, 2013.
- [26] A Gargallo-Peiró, X Roca, J Peraire, and J Sarrate. Distortion and quality measures for validating and generating high-order tetrahedral meshes. *Eng Comput*, in press.
- [27] X. Roca, J. Sarrate, and A. Huerta. Surface mesh projection for hexahedral mesh generation by sweeping. In *Proc. 13th Int. Meshing Roundtable*, pages 169–179, 2004.
- [28] X Roca, J Sarrate, and A Huerta. Mesh projection between parametric surfaces. *Commun. Numer. Met. Eng.*, 22(6):591–603, 2006.
- [29] P M Knupp. Algebraic mesh quality metrics. *SIAM J Sci Comput*, 23(1):193–218, 2001.
- [30] P M Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elem Anal Des*, 39(3):217–241, 2003.
- [31] P M Knupp. A method for hexahedral mesh shape optimization. *Int J Numer Meth Eng*, 58(2):319–332, 2003.
- [32] P M Knupp. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I - A framework for surface mesh optimization. *Int J Numer Meth Eng*, 48:401–420, 2000.
- [33] P M Knupp. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II - A framework for volume mesh

- optimization and the condition number of the Jacobian matrix. *Int J Numer Meth Eng*, 48:1165–1185, 2000.
- [34] L A Freitag and P Plassmann. Local optimization-based simplicial mesh untangling and improvement. *Int J Numer Meth Eng*, 49:109–125, 2000.
 - [35] L A Freitag and P M Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *Int J Numer Meth Eng*, 53:1377–1391, 2002.
 - [36] S P Sastry, S M Shontz, and S A Vavasis. A log-barrier method for mesh quality improvement and untangling. *Engineering with Computers*, pages 1–15, 2012.
 - [37] S P Sastry and S M Shontz. A parallel log-barrier method for mesh quality improvement and untangling. *Engineering with Computers*, pages 1–13, 2014.
 - [38] J M Escobar, G Montero, R Montenegro, and E Rodríguez. An algebraic method for smoothing surface triangulations on a local parametric space. *Int J Numer Meth Eng*, 66(4):740–760, 2006.
 - [39] J M Escobar, E Rodríguez, R Montenegro, G Montero, and J M González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Comput Methods Appl Mech Engrg*, 192(25):2775–2787, 2003.
 - [40] J M Escobar, R Montenegro, E Rodríguez, and G Montero. Simultaneous aligning and smoothing of surface triangulations. *Eng Comput*, 27(27):17–29, 2011.
 - [41] P J Frey and H Borouchaki. Geometric surface mesh optimization. *Comput Visual Sci*, 1(3):113–121, 1998.
 - [42] X Jiao, D Wang, and H Zha. Simple and effective variational optimization of surface and volume triangulations. *Eng Comput*, 27(27):81–94, 2011.
 - [43] Y Zhang, C Bajaj, and G Xu. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. *Commun numer meth eng*, 25(1):1–18, 2009.
 - [44] R Garimella and M Shashkov. Polygonal surface mesh optimization. *Eng Comput*, 20(20):265–272, 2004.
 - [45] R Garimella, M Shashkov, and P M Knupp. Triangular and quadrilateral surface mesh quality optimization using local parametrization. *Comput Methods Appl Mech Engrg*, 193(193):913–928, 2004.
 - [46] K Shivanna, N Grosland, and V Magnotta. An Analytical Framework for Quadrilateral Surface Mesh Improvement With an Underlying Triangulated Surface Definition. In *Proc. 19th Int. Meshing Roundtable*, pages 85–102, 2010.
 - [47] A Gargallo-Peiró, X Roca, J Peraire, and J Sarrate. Defining quality measures for meshes on parameterized cad surfaces. In *Proc. 21st Int. Meshing Roundtable*, 2012.
 - [48] J Yin and C Teodosiu. Constrained mesh optimization on boundary. *Eng Comput*, 24(3):231–240, 2008.
 - [49] A Kelly, L Kaczmarczyk, and C J Pearce. Mesh Improvement Methodology for 3D Volumes with Non-Planar Surfaces. In *Proc. 21st Int. Meshing Roundtable*, 2011.
 - [50] L Liu, Y Zhang, T J R Hughes, M A Scott, and T W Sederberg. Volumetric t-spline construction using boolean operations. *Engineering with Computers*, pages 1–15, 2013.
 - [51] OpenCASCADE. Open CASCADE Technology, 3D modeling & numerical simulation. <http://www.opencascade.org>, 2013.
 - [52] T Warburton. An explicit construction of interpolation nodes on the simplex. *Journal of engineering mathematics*, 56(3):247–262, 2006.
 - [53] A Gargallo-Peiró. *Validation and generation of curved meshes for high-order unstructured methods*. PhD thesis, Facultat de Matemàtiques, Universitat Politècnica de Catalunya, July 2014.
 - [54] T J Wilson. Simultaneous untangling and smoothing of tetrahedral meshes. Master’s thesis, ETSE Camins Canals i Ports de Barcelona, Universitat Politècnica de Catalunya, June 2011.
 - [55] J Nocedal and S Wright. *Numerical optimization*. Springer Verlag, 1999.